



King's Research Portal

Document Version

Publisher's PDF, also known as Version of record

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. *Educational Technology & Society*, 19(3), 38-46.
http://www.ifets.info/journals/19_3/5.pdf

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Arguing for Computer Science in the School Curriculum

Andrew Fluck^{1*}, Mary Webb², Margaret Cox², Charoula Angeli³, Joyce Malyn-Smith⁴, Joke Voogt⁵ and Jason Zagami⁶

¹University of Tasmania, Australia // ²King's College London, UK // ³University of Cyprus, Nicosia, Cyprus //

⁴Education Development Center, USA // ⁵University of Amsterdam, Windesheim University of Applied Sciences,

The Netherlands // ⁶Griffith University, Australia // Andrew.Fluck@utas.edu.au // mary.webb@kcl.ac.uk //

mj.cox@kcl.ac.uk // cangeli@ucy.ac.cy // jmsmith@edc.org // J.M.Voogt@uva.nl // j.zagami@griffith.edu.au

*Corresponding author

ABSTRACT

Computer science has been a discipline for some years, and its position in the school curriculum has been contested differently in several countries. This paper looks at its role in three countries to illustrate these differences. A reconsideration of computer science as a separate subject both in primary and secondary education is suggested. At EDUsumMIT 2015 it was argued that the major rationales for including computer science as a subject in the K-12 curriculum are economic, social and cultural. The paper explores these three rationales and also a beneficence matrix to assist curriculum designers. It also argues computer science is rapidly becoming critical for generating new knowledge, and should be taught as a distinct subject or content area, especially in secondary schools. The paper concludes by looking at some of the key questions to be considered when implementing computer science in the school curriculum, and at ways its role might change in the future.

Keywords

Computer science, Curriculum, K-12, Rationale, Primary school, Secondary school

Introduction

This paper looks briefly at the positioning of computer science in the curricula of several countries, and discusses the reasons for varying levels of priority from place to place and time to time. These reasons are then examined holistically to come to some broader sense of how and why the subject can be understood in context. From these reasons an argument is constructed to provide a rationale for including computer science in school curricula that can be tuned to fit particular contexts. Two ideas are critical to this paper: the contexts for school curricula considered in this article, and the definition of computer science. The school curricula considered in this article are primary and secondary schools, typically providing education for young people from the age of 5 to 18 years.

Computer science emerged from Babbage's Analytical Engine with Ada Lovelace's programming (to 1871) and from mathematics with Gödel's incompleteness theorem (1931). Its importance as a discipline became more widely accepted during the Second World War. The first university course in computer science was established at Cambridge (UK) in 1953, and the first university department of computer science at Purdue University (USA) in 1962. Subsequently, the growth of information technology has percolated through business, education and almost every aspect of society. This paper argues for computer science to become more widely taught in schools.

Computational thinking (Wing, 2006) is often seen as a key component of computer science in school curricula, and its importance was elaborated at EduSummit 2013 (Voogt et al., 2015). Definitions and arguments were made for its inclusion in school education. However, computational thinking is generally accepted as something which can be developed in computer science and which should also be integrated across the curriculum. In so doing, computational thinking is likely to transform what is taught in other subject areas (Fluck, 2003). We argue here that computer science should be formally taught in schools and that computational thinking is just one element of computer science, albeit an important one.

What is computer science?

Computer science is the term used in this article for a particular identified subject in the school curriculum. While "subject" is a common term, this varies from one country to another, so we use the word here to mean a particular

element of a school timetable, or a section of a regular report to stakeholders on a student's learning achievements. A subject can also be referred to as a domain or field of study, a discipline or a content area in the school curriculum.

Different names are given to around the world. Some alternatives are: informatics, digital technologies, or computing. Not all of these cover the same ground, but there is a great deal of commonality between them. The key focus for this article is on computer science (which we use for consistency throughout) as a separate subject distinct from the use of computers to support learning in other areas of the curriculum. As a separate subject, computer science has its own curriculum documents defining what students will know, the skills they will master and the attitudes they will acquire. Initially in this article we are concerned with this content rather than the way it is delivered. Later we will discuss approaches to implementation and delivery. The content will probably be assessed and achievement reported to stakeholders as an identified separate subject and may be taught in identified lessons in the school timetable but teaching arrangements may vary depending on age and other factors.

By comparison, there are general uses of computers in schools which are excluded from consideration here. Administrative uses (for recording attendance, managing resources etc.) are clearly excluded. Teacher construction of digital learning materials such as visual presentations (powerpoint slides) and other uses of office suite software are generally excluded, as is student word processing and similar activities. In particular, the general use of computers to support learning across the curriculum, often known as ICT (information and communication technology) is distinct from "computer science."

In some countries both ICT and computer science are taught side by side. In the United Kingdom, the "ICT curriculum" was phased out before "computing" with a strong foundation of computer science, was introduced as a curriculum subject.

Here are some popular definitions of computer science:

- It seeks to answer the following questions: What is information? What is computation? How does computation expand what we know? How does computation limit what we can know? (Denning, 2007).
- The study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society (Seehorn, Carey, Fuschetto, Lee, Moix, O'Grady-Cunniff, Owens, Stephenson & Verno, 2011).
- The scientific and practical approach to computation and its applications. It is the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information (Wikipedia, 2015).

In moving forward we favour the definition from Wikipedia, reflecting a more recent consensus. Disciplines are recognised by six features; terminology, non-routine intellectual work, axioms, openness, a body of knowledge, and identified sections of universities (Fein, 1959, p. 11). Chief amongst the axiomatic foundations of computer science as a discipline are the undecidability theorems (Gödel, 1931); the Turing–Church thesis (Turing, 1936); information theory (Shannon, 1948); and more recently, the empirical Moore's Law (1965). These theorems and their intellectual consequences have had widespread application in many other areas of intellectual endeavour. As we shall demonstrate later in this article, computer science is rapidly becoming crucial for progress in many other areas.

How are curricula determined?

Curriculum design is interwoven with pedagogy, and considered briefly here in order to position this current discussion within a broader context of theory and research. Key curriculum theorists such as Dewey (epistemology and personal growth), Popham (behavioural objectives), Foucault (power-knowledge), Vygotsky (social constructivism), and Bruner (psycho-cultural learning) have described the overt and hidden curriculum, and the aims, subject matter, methods and evaluation of learning. However, as in many other areas, theory and practical implementation are often in tension (Smith, 1973). Curriculum design is often a contested process, blending together what we know about educational maturation and pedagogy, social trends and political machinations tied to voting aspirations. Some view schools as resistant to change (Cuban & Tyack, 1995, p.7) and others have charted teacher change fatigue (Dilkes, Cunningham & Gray, 2014). Students spend up to 14 years in school, but curricula can be re-designed almost every five years, often with little regard to a teach-out or any phasing-in process. Any curriculum

has a political dimension, as well as technical and professional dimensions (Stenhouse, 1975). Depending upon national governance structures, education systems can be centralised or decentralised and the curricula they use can be designed top-down or bottom-up (Braslavsky, 2003).

Several drivers of curriculum development can be identified in the literature. Standards and accountability have generated a great deal of discussion (Taubman, 2009; Slattery, 2013). Entitlement is another driver, attempting to identify what a student needs to know and master in order to function in society (Young, 2013). Devising a rationale for computer science in school education requires an approach which can take account of how this range of issues and drivers may play out in different curriculum contexts. In addition to these more general considerations a major tension identified in relation to Computer Science is between the rapidity of technological change for computers and the general tendency of schools or curricula to respond slowly to such changes.

Computer science in different countries

We now look at three countries with differing trajectories of computer science curriculum development in schools. Cyprus has had such a subject in schools since the 1980s; the United Kingdom made a significant shift from ICT to computer science in 2014; and Australia recently endorsed its curriculum document for the subject in late 2015.

Cyprus is one of twelve European countries with computer programming and coding (as parts of computer science) as part of the curriculum as revealed by a survey for the European Schoolnet (Balanskat & Engelhardt, 2014). Computer science (in Cyprus) has been taught as a separate school subject in the secondary school grades 10-12 since 1987 and in grades 7-9 from 2001-2003. There is no distinct computer science subject in the primary school curriculum, but computers support other learning. The content of the course focuses on algorithmic thinking and programming which is compulsory for students aged 13-16. Older students can take it as an elective. Scratch and Alice are used to introduce the development cycle using drag and drop programming. Students compare the efficacy of different algorithms. Knowledge and achievement are assessed “through day-to-day assessment exercises, written assessment exercises and examinations” (Balanskat & Engelhardt, 2014, p. 19). All teachers of computer science in secondary education are required to be computer science graduates or graduates of a computing-related field.

The United Kingdom reviewed education in computing in schools from 2009-2012, culminating in a report by Furber (2012). This concluded that the positioning of ICT, as a subject, in the National Curriculum was counter-productive, and it was discontinued in 2012. In 2014 a new subject called “Computing” was introduced for students aged 5-16 (Department for Education, 2013). This encompasses computer science, information technology and digital literacy (Berry, 2013, p. 5). Students are expected to understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation; to analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems; evaluate and apply information; and be responsible, competent, confident and creative users of information and communication technology, including new or unfamiliar technologies, analytically to solve problems (Berry, 2013). The curriculum specification is quite brief at three pages but assessment approaches and teaching methods are provided as advice to teachers by the subject associations (see for example, Berry, 2013). New entrants to computer science teaching are expected to demonstrate their knowledge of the subject, and at secondary level they would normally be graduates of computer science or a computing-related field. An extensive professional development program has been developed for existing “ICT” teachers to equip them for the new curriculum.

Australia is a federated country, where responsibility for education was devolved to the eight states and territories in 1901. A statutory body was established in 2009 to frame a national curriculum which began implementation in 2013 with four subjects and ICT as a general capability to be taught across and in support of the other subject areas. In November 2015 “Digital Technologies” was endorsed as an additional subject in the Technologies area, and will be gradually introduced into schools region by region, with reports on student achievement being sent to stakeholders in about 2018. Digital technologies looks at digital system components and data representation; and students learn to collect, manage and analyse data, and to create digital solutions. Computational thinking is key to the syllabus, encompassing abstraction, data collection, representation and interpretation, specification, algorithms and implementation. Methods of teaching or assessment are unspecified, but content is described and achievement standards have been set at five age levels for students aged from about 5 to 16 years old. The subject will be compulsory from about age 6 to age 14, and elective thereafter. Teacher professional development will be conducted

by individual states and territories, but central resources such as the learning object repository SCOOTLE will probably be useful.

Given the different pathways computer science has taken in these three countries, it seems timely to examine the arguments for its inclusion as a separate subject in school curricula.

The arguments for computer science in the curriculum

At EDUsummIT 2015 it was argued that the major rationales for including Computer Science as a subject in the K-12 curriculum are economic, social and cultural, partly echoing Hawkrigge's (1990) justification for using computers in schools.

The economic rationale is a strong driver for inclusion of computer science in the curriculum. The economic rationale rests not only on the need for a country to produce computer scientists to sustain a competitive edge in a world driven by technology, but also on the requirement for computer science-enabled professionals in all industries to support innovation and development. World trade in information technology products and services continues to grow. Worldwide exports of computer and information services grew between nine and 32 percent from 2005 to 2015 and imports grew from 6 to 28% in the same period (World Trade Organization, 2014a, p. 134). Computer and information services were identified as the seventh largest trade sector by value, particularly between developed countries, from 2008 to 2013 (UNCTAD, 2014, p. 7). Export growth in the sector was up to 20% in the period 2011-13 (UNCTAD, 2014, p. 11), and over the period 2005-13 was the highest of any global trade sector (WTO, 2014b, p. 25). Both Australia and Europe forecast shortages of computer scientists of 100 to 500 thousand (ACS, 2015, p. 34; Husing & Korte, 2010). Businesses in the sector compete for staff and intellectual property development against other sectors. Alongside these growth trends must be placed the increasing stress in trade negotiations on intellectual property rights and the trend for software to be protected by both copyright and patent legislation. Recent trade agreements elongate intellectual property ownership terms for up to 70 years from the death of the author (AU FTA 2013). This underlines the economic importance of computing professionals and the intellectual property they produce. The economic rationale is therefore compelling for governments devising national curricula, and who see future trade prospects dependent upon computers. Computers have also made previously un-tradable services such as banking become more globally mobile. This makes sourcing local talent more crucial in controlling and building the local national economy.

This leads to the social rationale for computer science. The social rationale emphasises the value in society of active creators and producers rather than passive consumers of technology. Such capability provides people with power to lead, create and innovate within society and therefore is also an issue of entitlement to "powerful knowledge" (Young, 2013) giving individuals opportunities to choose their role in society. Since so much of our lives depends upon computers, surely students should become confident in their use? The social rationale goes further, and discusses the impact of technologies on societies. Computing makes possible new forms of governance (distributed daily or enhanced democracy) and redefines our sense of privacy (Tsohou, Lee, & Irani 2014). Robots comfort senior citizens in senior care institutions (Sharkey & Sharkey, 2012), and taxis or trucks cease to need drivers (Ticoll, 2015, p. 48). Technology can change our sense of ethical behaviour. Rather than being oppressed by innovation shock, a society equipped with its own creative proponents of new ideas is more likely to sift them and control their impact. This is a strong argument for teaching computer science in schools, making it possible to generate new content and applications locally, and enhancing the agility of each nation to deal with inevitable changes in the world. Competition for resources with other nations is just one such change pressure, so falling behind is not a desirable option. In some ways the social rationale overlaps the economic rationale, but the emphasis is on attitudinal aspects rather than knowledge or skill.

The cultural rationale for computer science in schools looks at the integrity of local values and customs. The cultural rationale enables people to be drivers of cultural change, rather than having change imposed or mediated through technological developments from outside their society (Webb et al., 2015, p.61). For example, Hollywood produces many movies every year which are successfully exported all over the world. Embedded in these films are language, customs, attitudes, ethical values and mores which reflect the USA context. Surprisingly, these attributes are not universally shared, and hence other centres of production have emerged, for example, Bollywood, which made twice as many films in 2011 (Matusitz & Payano, 2011). Nollywood, films made in or for Nigeria and Africa, is also

developing (Oyeniya, 2013). Given computer games now outsell the film industry, there is additional reason for local creative talent to provide competition for imported embedded values and language. In 2014 the global video game market was estimated to be worth \$US102 billion, compared to global box office receipts for films of \$US36 billion (van der Meulen, 2013; MPAA, 2014, p. 2). The local talent pool becomes more imperative when we take into account the greater cognitive engagement in computer games, and therefore their greater capacity to transmit culture. In addition, social media are connecting groups of people across the globe in personal ways almost instantaneously. The social and semantic webs facilitate data creation, contribution and sharing on an unprecedented scale. Online shopping is changing the structure of production and distribution of goods. Friendships are made online (Ofcom, 2008). Therefore governments are motivated to ensure national cultural values are preserved through locally educated computer science professionals participating in the arts and related fields.

Alongside these three rationales, we can frame the inclusion of computer science in school curricula in a broader context in order to give curriculum designers greater scope to determine its feasibility and desirability in schools curricula. We suggest two dimensions for evaluating the contribution of computer science. The first dimension is the beneficial context: the individual learner; the society in which they live; humanity and the ecology upon which we all depend; and the wider universe. The other dimension concerns the period for the benefits of the learning to be experienced: immediately; the lifetime of the individual learner; years within a social system transformation; the expected duration of humanity; or the lifetime of the universe.

These dimensions are deliberately drawn large to ensure we do not omit any potential benefit from the proposed learning.

We also posit computer science as necessary for education because of its increasing importance for knowledge generation in many areas of human endeavour. Depriving school students of the terminology, the axioms and other components of the discipline would impair their capacity to understand or contribute to this new knowledge. Examples include the Nobel prizes for chemistry in 1998 and 2013 (Nobelprize.org, 2015) where computers were fundamental; astro-physics looks at information flows in black holes (Braunstein et al., 2013); data mining helps solve problems in medicine (Yavlinsky, 2015) and parallels have been drawn between the entropies of energy and of information in the universe (Avery, 2003; Shannon, 1948, p. 11). Not only is computer science heralding new developments in chemistry, physics and biology, but data science is providing new methods for knowledge discovery. These new methods of Data Science compare with the Scientific Method as shown in Table 1.

Table 1. Comparison of scientific method with data science method

Scientific method	Data science method
1. Observe phenomena	1. Become aware of a problem
2. Propose a hypothesis	2. Identify data sources
3. Design an experiment as a fair test of the hypothesis	3. Propose an automated analysis process
4. Collect data, analyse results	4. Test and validate analysis process
5. Confirm or modify the hypothesis, start over	5. Share results, start over

Data science links to machine learning to make new discoveries (Phoboo, 2014; Adam-Bourdarios et al., 2015, p.27) and lies at the heart of many recent global businesses such as eBay, Amazon, Facebook, Twitter and Spotify (Gershkoff, 2015). Therefore the link between computer science and innovation makes a strong case for inclusion in the school curriculum.

There are other immediate benefits to students learning computer science. Some examples illustrate these benefits which strengthen the argument:

Coding is about thinking. Putting a process into a particular code (writing a program for a computer) requires precision. This is analogous to the precision required in literacy skills of writing, but while people can understand intent or infer spelling, computers are far more literal. Therefore a child skilled at coding, may by transference, be more precise in their thought and have greater capacity to communicate. The beneficence is immediate and personal to the learner.

Generally, with modern personal computers, students can get fairly immediate feedback on the accuracy of their initial coding attempts. Naturally, in the learning process, some early efforts result in less than complete success – or

failure. Handled well within the classroom, this can be an opportunity to build resilience. The nature of the failure (perhaps printing out all but the last number in a sequence) can indicate the corrective editing needed to achieve success. This is an important educative step of understanding for many students. In this case the beneficence may not be so immediate, and affects both the learner and the society in which they live.

The strength of the arguments for including computer science into school curricula will vary from nation to nation, and time to time. In the second decade of the twenty-first century, there are strong economic and social reasons for its adoption, culminating in an entitlement perspective benefiting both the individual and the society in which they live.

Implementation of computer science as a subject

Literacy and numeracy are often woven into many subject areas of the school curriculum, creating themes which are often the focus for much primary (elementary) schooling. Therefore there is a suggestion that computer science could be integrated into other thematic programs of work, particularly in this primary sector. After all, most curriculum specification documents identify what should be learned, not how it should be organised and taught. Arguments for the integration of computer science need to be looked at in context. If teachers are knowledgeable, confident, fully trained in computer science and have adequate resources, it would be very difficult to mount an argument against integration. However, for those countries introducing or re-introducing the subject into school curricula, an integration policy could mask the extent to which the new content is being mastered by students. A large scale study of 37 countries (Plomp et al., 2009) concluded that integration of ICT use into other subjects was spasmodic and ineffective. Voogt and ten Brummelhuis (2014) illustrated how information literacy in the Netherlands disappeared as a separate subject, because of poorly trained teachers and a vague place in the curriculum. These examples indicate computer science could have a similar fate if only taught through integration.

There are several challenges for schools where computer science is (re-)introduced. New pedagogical content knowledge may be required, as well as teaching resources such as programming development environments and textbooks or their equivalents. While computers themselves are getting less expensive, access and particularly individual access, cannot be assumed. A great deal of computer science can be taught “unplugged” (Bell & Newton, 2013) but greater alignment with the three main rationales is likely to be greater if computers or BYOT (bring your own technology) like smartphones are available.

Another tension is the rapid rate at which innovations are changing our understanding of computer science. To become entrenched in schools, the subject needs clear specifications for learning outcomes, assessment and standards which go beyond the tendency to have the curriculum build around technology innovations instead of a deep understanding of computer science (Webb et al., 2015, p.64). Whilst we may need to live with a continually renegotiated curriculum (Webb, 2014), this is not easy to communicate and interpret between the policy level of education systems and implementation in the classroom. France is planning an integrated approach (Balanskat & Engelhardt, 2014, p.17) so will be worth monitoring.

Several other challenges and solutions were considered at EduSummit 2015 (Webb et al., 2015), which made recommendations for policy makers, educators, industry partners and researchers.

Into the future

Many things in education oscillate in the style of a pendulum. Since the year 2000 computer science has been less popular, but in 2015 it is becoming resurgent. Several countries are re-introducing computer science (under this or a similar subject title) from 2015 onwards. Australia, USA, UK, have committed or initiated computer science as a subject, whilst the Czech Republic, Denmark, Lithuania, Poland and the Netherlands are in the process of doing so, mostly from the first classes of primary school.

This brings us to some of the more important questions educators need to answer with respect to computer science as a separate subject in the school curriculum. Some of these questions were raised at EduSummit 2015 and many of

them interact with one another. Winch's (2013) concept of "epistemic ascent" is key to understanding how these questions can be used to frame computer science curriculum design:

- What is the range of skills and understanding that should be developed in computer science?
- Are such skills and understanding necessary for everyone? Should it be and remain compulsory?
- At what age should computer science education commence?
- How many computing languages or frameworks should a student be exposed to in the span of schooling from K-12?
- How varied should these languages be? Should a variety of paradigms be explored?
- How closely should the curriculum match computers available to schools and students?
- What consideration in curriculum design should be given to emerging technologies such as quantum networks and computing?
- What pedagogical approaches are likely to be appropriate, and how do they vary with age and other factors?

Finally, to bring together some of the issues briefly mentioned in this paper, it may be worth making a case study from quantum computing. Let us assume all the above questions have been answered in a particular national context, and a clear specification for computer science in the curriculum has been defined and implemented. Teachers have been trained and assessments of achievement show targeted students are largely attaining the intended learning outcomes. Many current curricula focus on procedural or algorithmic thinking when devising digital solutions. However, this may not be appropriate with emerging quantum computers where the fundamental components are qubits which can be in a probabilistic superposition of several states simultaneously. How can our agile curriculum design accommodate such a radical departure from established content knowledge? Touted applications for quantum computers include cryptanalysis, typically by factoring very large numbers using the Shor algorithm (1997). The essential approach is to try all possible solutions simultaneously. This has parallels in the declarative programming paradigm where code specifies what computation should be performed, and not how to compute it. Prolog was once a typical declarative language, and functional languages such as Haskell may be similarly suitable. This makes an argument for computer science curricula to have an eye to the future, with content chosen to some extent with awareness of emerging technologies.

References

- Australian Computer Society [ACS]. (2015). *Australia's digital pulse: Key challenges for our nation – digital skills, jobs and education*. Sydney, Australia: Deloitte Access Economics.
- Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kégl, B., & Rousseau, D. (2015). The Higgs Boson machine learning challenge. *Journal of Machine Learning Research: Workshop on High-energy Physics and Machine Learning and conference proceedings*, 42, 19-55. Retrieved from <http://jmlr.org/proceedings/papers/v42/cowal4.pdf>
- Avery, J. (2003). *Information theory and evolution*. Singapore: World Scientific Publishing.
- Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussels, Belgium: European Schoolnet. Retrieved from http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887
- Bell, T., & Newton, H. (2013). Unplugging computer science. In D. M. Kadijevich, C. Angeli & C. Schulte (Eds.), *Improving computer science education* (pp. 52-65). New York, NY: Routledge.
- Berry, M. (2013). *Computing in the national curriculum: A Guide for primary teachers*. Bedford, UK: NAACE – Computing at School.
- Braslavsky, C. (2003). *The Curriculum*. Geneva, Switzerland: UNESCO international bureau of education. Retrieved from http://portal.unesco.org/education/en/file_download.php/3c70bf409f9a7d3e072f80d8c7dcc597cecilia+e.pdf
- Braunstein, S. L., Pirandola S., & Życzkowski, K. (2013). Better late than never: Information retrieval from black holes. *Physical Review Letters*, 110, 101301.
- Cuban, T., & Tyack, D. B. (1995). *Tinkering toward Utopia: A Century of public school reform*. Cambridge, MA: Harvard University Press.
- Denning, P. J. (2007). Computing is a Natural Science. *ACM Communication*, 50(7), 13-18.

- Department for Education. (2013). *National curriculum in England: Computing programmes of study*. London, UK: Department for Education.
- Dilkes, J., Cunningham, C., & Gray, J. (2014). The New Australian curriculum, teachers and change fatigue. *Australian Journal of Teacher Education*, 39(11), 45-64.
- Fein, L. (1959). The Role of the university in computers, data processing, and related fields. *Communications of the ACM*, 2(9), 7-14.
- Fluck, A. E. (2003). *Integration or transformation? A Cross-national study of information and communication technology in school education* (Unpublished doctoral dissertation). University of Tasmania, Australia. Retrieved from <http://eprints.utas.edu.au/232/>
- Furber, S. (2012). *Shut down or restart? The Way forward for computing in UK schools*. London, UK: The Royal Society.
- Gershkoff, A. (2015). *How to stem the global shortage of data scientists*. TechCrunch newsletter. Retrieved from <http://techcrunch.com/2015/12/31/how-to-stem-the-global-shortage-of-data-scientists>
- Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. [On formally undecidable propositions of Principia Mathematica and related systems I]. *Monatshefte für Mathematik und Physik*, 38, 173-98.
- Hawkrige, D. (1990). Who needs computers in schools, and why? *Computers & Education*, 15(1-3), 1-6.
- Husing, T., & Korte, W. B. (2010). *Evaluation of the implementation of the communication of the European Commission: E-skills for the 21st century*. Bonn, Germany: Empirica. Retrieved from <http://hdl.voced.edu.au/10707/323186>
- Matusitz, J., & Payano, P. (2011). The Bollywood in Indian and American perceptions: A Comparative analysis. *India Quarterly: A Journal of International Affairs*, 67(1), 65-77. doi:10.1177/097492841006700105
- Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8), 114-117.
- Motion Picture Association of America [MPAA]. (2014). *Theatrical market statistics 2014*. Washington, DC: MPAA. Retrieved from <http://www.mpa.org/wp-content/uploads/2015/03/MPAA-Theatrical-Market-Statistics-2014.pdf>
- Nobelprize.org (2015). *All nobel prizes in Chemistry*. Retrieved from http://www.nobelprize.org/nobel_prizes/chemistry/laureates/
- Office of Communications [Ofcom] (2008). *Social networking: A Quantitative and qualitative research report into attitudes, behaviours and use*. London, UK: Ofcom. Retrieved from <http://stakeholders.ofcom.org.uk/binaries/research/media-literacy/report1.pdf>
- Oyeniya, A. (2013, November 8). *Nollywood at 20? The Punch Newspaper*. The Punch NG.
- Phoboo, A. E. (2014). *Machine learning wins the Higgs challenge*. Retrieved from <https://cds.cern.ch/record/1972036?ln=en>
- Plomp, T., Anderson, R.E., Law, N., & Quale, A. (Eds.) (2009). *Cross-national information and communication technology: Policies and practices in education*. Charlotte, NC: Information Age Publishing.
- Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B. B., Stephenson, C., & Verno, A. (2011). *CSTA K-12 computer science standards: Revised 2011*. New York, NY: Computer Science Teachers Association (CSTA) and the Association for Computing Machinery, Inc (ACM).
- Shannon, C. E. (1948). A Mathematical theory of communication. *Bell System Technical Journal*, 27, 379-423, 623-656.
- Sharkey, A., & Sharkey, N. (2012). Granny and the robots: Ethical issues in robot care for the elderly. *Ethics and information technology*, 14(1), 27-40. doi:10.1007/s10676-010-9234-6
- Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Society for Industrial and Applied Mathematics Journal on Computing*, 26(5), 1484-1509. doi:10.1137/S0097539795293172
- Smith, T. B. (1973). The Policy implementation process. *Policy Sciences*, 4, 197-209.
- Slatery, P. (2013). *Curriculum development in the postmodern era: Teaching and learning in an age of accountability* (3rd ed.). Abingdon, UK: Routledge.
- Stenhouse, L. (1975). *An Introduction to curriculum research and development*. London, UK: Heinemann.
- Taubman, P. M. (2009). *Teaching by numbers: Deconstructing the discourse of standards and accountability in education*. Upper Saddle River, NJ: Pearson Merrill/Prentice-Hall.

- Ticoll, D. (2015, October). *Driving changes: Automated vehicles in Toronto*. Discussion paper from University of Toronto Transportation Research Institute. Retrieved from [https://www1.toronto.ca/City%20Of%20Toronto/Transportation%20Services/TS%20Publications/Reports/Driving%20Changes%20Final%20\(compressed\).pdf](https://www1.toronto.ca/City%20Of%20Toronto/Transportation%20Services/TS%20Publications/Reports/Driving%20Changes%20Final%20(compressed).pdf)
- Tsohou, A., Lee, H., & Irani, Z. (2014). Innovative public governance through cloud computing: Information privacy, business models and performance measurement challenges. *Transforming Government: People, Process and Policy*, 8(2), 251-282. doi:10.1108/TG-09-2013-0033
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungs problem. In *Proceedings of the London Mathematical Society* (Vol. 2, No. 42, pp. 230-265). Retrieved from https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf
- United Nations Conference on Trade and Development [UNCTAD]. (2014). *Key statistics and trends in international trade 2014*. Geneva, Switzerland: United Nations Publication. Retrieved from http://unctad.org/en/PublicationsLibrary/ditctab2014d2_en.pdf
- van der Meulen, R. (2013, October). *Gartner says worldwide video game market to total \$93 billion in 2013* [Web log post]. Retrieved from <http://www.gartner.com/newsroom/id/2614915>
- Voogt, J., & ten Brummelhuis, A. (2014). Information literacy in the Netherlands Information literacy in the Netherlands: Rise, fall and revival. In A. Tatnall & B. Davey (Eds.), *Reflections of the history of computers in education* (pp. 83-92). New York, NY: Springer.
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728.
- Webb, M. (2014). Considerations for the design of computing curricula. In T. Brinda et al. (Eds.), *Commentarii Informaticae Didacticae, KEYCIT- Key competencies in informatics and ICT* (pp. 267-284). Potsdam, Germany: Universitätsverlag Potsdam.
- Webb, M., Fluck, A., Cox, M., Angeli-Valanides, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2015). Curriculum: Advancing understanding of the roles of computer science/informatics in the curriculum. In K. W. Lai (Ed.), *EDUsummit 2015 Summary Report* (pp. 61-70). Retrieved from <http://www.curtin.edu.au/edusummit/local/docs/edusummit2015-ebook.pdf>
- Wikipedia. (2015). *Computer science*. In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Computer_science
- Winch, C. (2013). Curriculum design and epistemic ascent. *Journal of Philosophy of Education*, 47(1), 128-146. doi:10.1111/1467-9752.12006
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- World Trade Organization [WTO] (2014a). *International trade statistics 2014*. Retrieved from https://www.wto.org/english/res_e/statis_e/its2014_e/its2014_e.pdf
- World Trade Organization [WTO] (2014b). *World trade report 2014*. Retrieved from https://www.wto.org/english/res_e/booksp_e/world_trade_report14_e.pdf
- Yavlinksy, A. (2015). *Memantic: A Medical knowledge discovery engine*. Retrieved from <http://arxiv.org/pdf/1503.05781v1.pdf>
- Young, M. (2013). Overcoming the crisis in curriculum theory: A Knowledge-based approach. *Journal of Curriculum Studies*, 45(2), 101-118.